



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA

Sistema de venta de contenidos digitales bajo demanda.

On demand digital content sales system.

Realizado por
Fernando Jiménez Iglesias

Tutorizado por
Antonio Maña Gómez

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JUNIO DE 2021

Fecha defensa:

Resumen

En este trabajo de fin de grado se presenta una aplicación web destinada a ser un Marketplace de videojuegos que ofrece sistema de precios dinámicos basado en un **algoritmo** que usa la tendencia de ventas que tiene el juego para reevaluar de forma periódica el precio del mismo. Para el desarrollo de la web se usa como base el CMS de **WordPress**, de forma que las principales tecnologías usadas son **PHP**, **MySQL** y **CSS** haciendo uso de diversos plugins para garantizar un mejor funcionamiento de la web y una reducción en el tiempo de desarrollo. Igualmente, para facilitar su adopción, el algoritmo desarrollado en este trabajo se implementa en forma de un **plugin de WordPress**.

El objetivo final es que el algoritmo asigne un **precio adaptativo** a los productos de nuestra web de forma que sea más atractiva para los usuarios en el mercado buscando **maximizar las ventas** y la **rentabilidad** por parte del vendedor.

Palabras clave: Plugin de WordPress, Algoritmo, WordPress, PHP, MySQL, CSS, precio adaptativo, maximizar las ventas y la rentabilidad.

Abstract

In this Bachelor Thesis we present a web application that materializes a video game marketplace offering a dynamic pricing system based on an algorithm that uses the games' sales trends to periodically reevaluate their price. WordPress CMS is used as the basis for the development of the website. Consequently the main technologies used are PHP, MySQL and CSS. We use of various plugins to ensure a better functioning of the web and a reduction in development time. Likewise, to facilitate its adoption, the algorithm developed in this work is also implemented in the form of a WordPress plugin.

The ultimate goal is that the algorithm assigns an adaptive price to the products on our website so that it is more attractive to users in the marketplace seeking to maximize sales and profitability on the part of the seller.

Keywords: Adaptive pricing, Sales and profitability, Algorithm, WordPress, WordPress Plugin, PHP, MySQL, CSS,.

Índice

RESUMEN	1
ABSTRACT	1
ÍNDICE.....	1
INTRODUCCIÓN	1
1.1 MOTIVACIÓN	1
1.2 1.2 OBJETIVOS	1
1.3 ESTRUCTURA DE LA MEMORIA.....	2
1.4 ESTADO DEL ARTE.....	2
1.5 TECNOLOGÍAS UTILIZADAS.....	3
ANÁLISIS	5
2.1 OBTENCIÓN DE REQUISITOS.....	5
2.1.1 <i>Requisitos funcionales</i>	5
2.1.2 <i>Requisitos no funcionales</i>	6
2.2 ACTORES	7
2.3 DIAGRAMAS DE CASOS DE USO.....	7
2.3.1 <i>Casos de Uso</i>	8
IMPLEMENTACIÓN DE LA WEB	13
3.1 VISIÓN GENERAL	13
3.2 WORDPRESS	13
3.3 HOSTING	18
3.4 PLUGINS.....	19
DISEÑO DEL ALGORITMO	20
4.1 VISIÓN GENERAL	20
4.2 DATA-SETS.....	20
4.3 PROTOTIPOS INICIALES	21
4.4 DESARROLLO DE LA APLICACIÓN DE CONSOLA.....	22
4.5 PLUGIN.	28
4.6 BASE DE DATOS DEL PLUGIN	30
4.7 FRONT-END PLUGIN.....	32
4.8 INSTALACIÓN DE UN SERVIDOR LOCAL (XAMPP)	33
4.9 PROBLEMAS DURANTE EL DESARROLLO	35
CONCLUSIONES.....	36
5.1 CONCLUSIONES.....	36
5.2 FUTURAS MEJORAS Y LÍNEAS DE INVESTIGACIÓN	36
REFERENCIAS	39
MANUAL DE USUARIO E INSTALACIÓN	41
6.1 INTRODUCCIÓN.....	41

6.2	INSTALACIÓN	41
6.3	USO DEL PLUGIN.....	42

1

Introducción

1.1 Motivación

Hoy en día, encontramos que los sistemas de compra y venta de contenidos digitales están bastante estandarizados. En particular, existen muchos contenidos digitales cuyo “valor” percibido evoluciona en el tiempo en función de diversos parámetros, como el interés que despierta, novedad, etc. y para los que un esquema de precios simple y rígido no se adapta bien y no maximiza los beneficios a los proveedores de dichos contenidos.

Un ejemplo lo tenemos en el mundo de la venta de videojuegos. Existen infinidad de webs que se dedican a ello, pero podríamos decir que casi todas ellas parten de la misma base a la hora de vender sus videojuegos, precio base y oferta de descuento. Este esquema rígido resulta negativo para muchos juegos. Es frecuente que algunos juegos no alcancen su potencial de ventas debido a que salen con un elevado precio, por lo que no llegan a ser tan populares como la competencia gratuita, y dado que necesitan un gran volumen de jugadores para sacar todo su potencial, se ven obligados a recurrir a bots para suplir la incapacidad de alcanzar su masa crítica, empeorando en ocasiones la experiencia del jugador.

Sin embargo, en estas webs no se aprovecha la posibilidad de variar el precio del producto de forma automática para en un momento que las ventas están en declive reformarlas con una bajada de precios o en caso de que el precio de salida fuera demasiado bajo aumentarlo en función del éxito del mismo.

1.2 1.2 Objetivos

Este trabajo de fin de grado tiene como objetivo la creación de un mercado online (Marketplace) basado en el CMS de WordPress que implementara un algoritmo específicamente desarrollado para adaptar el precio de los videojuegos en base a el número de compradores que está teniendo el mismo, pudiendo así optimizar las ventas de manera automática, en base a las circunstancias del mercado,

mejorando los beneficios y ofreciendo a los jugadores la oportunidad de adquirir el juego a un precio competitivo y que refleje el estado real del mismo.

El algoritmo se implementara de forma externa a la web a modo de plugin de modo que pueda adaptarse a la venta de otros tipos de contenidos digitales (por ejemplo, música, videos, etc.) cuya demanda evoluciona en el tiempo.

1.3 Estructura de la memoria

La memoria se ha organizado en base a las distintas partes del desarrollo del proyecto, ahora me dispongo a dar una breve explicación en los siguientes puntos:

Capítulo 1: Introducción. Breve prologo exponiendo la situación inicial y los motivos por los cuales damos comienzo a este proyecto.

Capítulo 2. Análisis. Durante este capítulo analizamos los distintas partes del proyecto mediante la identificación de los requisitos funcionales y no funcionales, aportando una breve descripción de los mismos y descubriendo así cuales son los actores implicados y los casos de uso a estudiar.

Capítulo 3. Implementación de la web. Implementación del sitio web en si a través de WordPress.

Capítulo 4: Diseño del algoritmo. Implementación y pruebas del algoritmo en sí, más su desarrollo como plugin de WordPress

Capítulo 5: Conclusiones. Conclusiones obtenidas a lo largo del proyecto y posibles mejoras para el desarrollo del plugin en el futuro.

Apéndice 6 A: Instalación y Manual de Usuario. Manual de uso de la web y el plugin en si mismo.

1.4 Estado del arte

En este punto vamos a evaluar las distintas opciones que ahora mismo existen en el mercado para la automatización precios y vamos a compararlas entre sí para ver que ideas podemos incluir en nuestro algoritmo.

Paarly

Es un software de origen francés que se dedica al análisis de precios y monitoreo de la competencia. La plataforma incluye tecnología de inteligencia artificial (IA), que permite a las organizaciones analizar las tendencias del mercado en tiempo real utilizando gráficos, tablas de comparación y hojas de cálculo.

Básicamente nos da un análisis continuo de la competencia, pero esto no es nuestro objetivo principal, ya que esto mejoraría las ventas pero nuestro objetivo es maximizar el beneficio.

Omnia-retail

Omnia's Dynamic Pricing es un software de optimización de precios para comercios minoristas. Los minoristas retoman el control sobre sus precios, ahorran tiempo y aumentan sus márgenes y ganancias.

Parece una solución bastante robusta pero como mejora queremos que nuestra solución sea más accesible que la suya para aquellos usuarios que tengan una web en WordPress ya que la implantación en su web de será de solo unos cuantos clics.

Reactev

Reactev es un producto diseñado exclusivamente para la venta en el sector retail (al por menor), cubre el ciclo completo del pricing y eCommerce. Utiliza el procesamiento por Big Data y técnicas de Deep Learning para predecir y reaccionar a los cambios del mercado.

Se podría decir que es que tiene un enfoque más parecido al nuestro pero Creemos que un buen algoritmo determinista puede ser tan buena solución como la que ellos proponen.

1.5 Tecnologías utilizadas

En este apartado explicaremos cuales son las principales herramientas, *frameworks*, lenguajes de programación y librerías usadas en el desarrollo e implementación del proyecto:

- Taiga: Taiga es la herramienta de gestión de proyectos para equipos multifuncionales ágiles. Ofrece una gran variedad de prestaciones y al mismo tiempo es muy simple de poner en marcha gracias a su intuitiva interfaz de usuario.
- Visual Studio Code: Visual Studio Code es una potente herramienta para la edición de código, fue desarrollado por Microsoft para las principales plataformas (Windows, Linux y macOS). Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto.
- XAMP: Una distribución de Apache que incluye varios software libres. Proporciona todo lo necesario para poder ejecutar un proyecto WordPress de forma local.[1]
- Visual Studio: Es un potente entorno de desarrollo software integrado para Windows y macOS. Es compatible con múltiples lenguajes de programación como C++, c#, VisualBasic, .NET, ...
- JavaScript: JavaScript es un lenguaje de programación interpretado orientado a objetos (OO). Normalmente se utiliza en los navegadores web y logra extender su funcionalidad permitiendo la creación con objetos que permiten a los scripts interactuar con el usuario, controlar el navegador web y modificar el contenido del de la propia web.
- Bootstrap: Bootstrap es un framework front-end utilizado para desarrollar aplicaciones web y sitios mobile first, es decir, es capaz de adaptarse a la pantalla del dispositivo utilizado por el usuario.
- GitHub: Una de las principales plataformas donde alojar proyectos de cualquier desarrollador.

- C#: Lenguaje de programación multiplataforma desarrollado por Microsoft como parte de su plataforma .NET.
- PHP: PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.
- MagicDraw: Es una herramienta CASE (Computer Aided Software Engineering) utilizada para realizar los diagramas de la aplicación usando la notación UML.

2

Análisis

2.1 Obtención de requisitos

Antes de proceder al análisis, comentar que en este apartado se especifican los requisitos funcionales y no funcionales de forma conjunta dentro del proyecto, es decir no hay una subdivisión previa en subproyectos o componentes para delimitar que parte del proyecto se encarga de que requisito, ya que esta división sería definida más adelante.

2.1.1 Requisitos funcionales

Estos son los requisitos más orientados a especificar las capacidades que los distintos usuarios pueden aplicar en el sistema.

Identificador	Nombre	Descripción
RF1	Configurar parámetros	Los usuarios deben de ser capaces de modificar y adaptar los criterios del algoritmo para adaptar su funcionamiento a lo que se desee.
RF2	Generar Gráficas	Contendrá una serie de gráficas para demostrar cómo evoluciona en el tiempo los precios.

RF3	Importar casos reales	Debemos de poder importar casos reales durante el periodo de testeo del algoritmo para poder ver su eficacia.
RF4	Comprar en la web	Debe de ser posible para el usuario la compra de los productos en la web.
RF5	Realizar comentarios	Los productos de la web pueden tener una lista de comentarios.
RF6	Buscar producto	la web debe de disponer de algún filtro para facilitar la busque de un producto en concreto.
RF7	CRUD de productos en la web	La web debe de permitarnos la creación , lectura, actualización y eliminación de productos.

2.1.2 Requisitos no funcionales

Ahora damos paso a aquellos requisitos no funcionales que debe de cumplir el sistema con independencia de los Actores dentro del sistema.

Identificador	Nombre	Descripción
RNF1	Evaluación de precio dinámica no supervisada	Una vez configurado el algoritmo para un producto en concreto el recalcule del precio no requerirá de ninguna acción por parte de los actores más a ya del volumen de ventas que este tenga.
RNF2	Ejecución automática / no inerencia	El algoritmo tiene que ser totalmente independiente del administrador una vez este haya sido

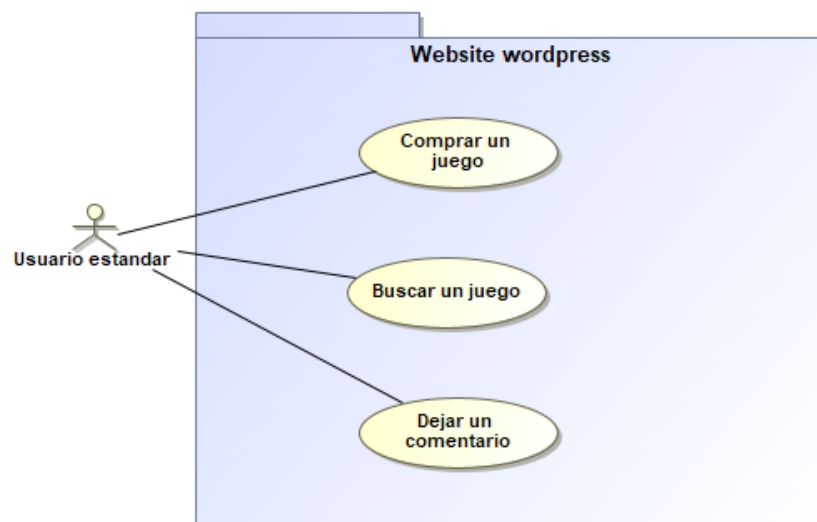
		configurado para un producto determinado.
RFN3	Sencillo de usar	El plugin deberá de tener una interfaz simple y clara de forma que sea lo bastante explicativo por si mismo y no requiera de guías para su uso.
RFN4	Web simple	La web no debe de tener demasiado exceso de elementos ya que por limitaciones del hosting gratuito el espacio que disponemos para la web es reducido.

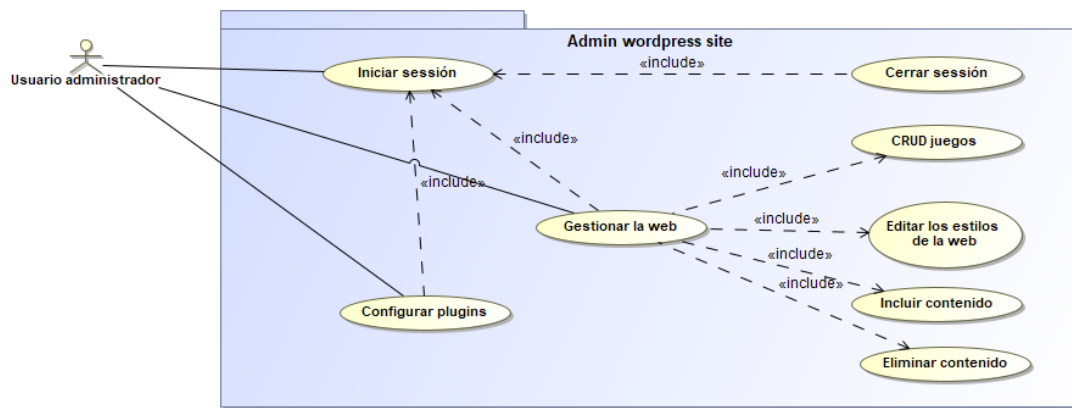
2.2 Actores

A continuación nos disponemos a explicar cuáles son los actores que disponemos dentro de nuestro sistema y que tipo de acciones desempeñan dentro del mismo:

- Usuario estándar: Representa a los potenciales compradores dentro del sistema.
- Usuario administrador: Es un usuario con todos los privilegios sobre la página de WordPress.

2.3 Diagramas de casos de uso.





2.4 Casos de Uso.

Nombre	Descripción
Caso de uso	Configurar parámetros.
Actores	Usuario administrador.
Descripción	Los usuarios deben de ser capaces de modificar y adaptar los criterios del algoritmo para adaptar su funcionamiento a lo que se desee.
Precondiciones	<ul style="list-style-type: none"> - El plugin debe de estar instalado en la web. - Deben de existir productos.
Escenario principal	<ol style="list-style-type: none"> 1.El administrador inicia sesión en el panel de administración de WordPress. 2.Selecciona un producto que dese configurar dentro de la pestaña de nuestro plugin. 3.El administrador introduce la configuración para el producto.
Escenario alternativo	No existe escenario alternativo.

Nombre	Descripción
Caso de uso	Generar Gráficas
Actores	Usuario administrador.
Descripción	Contendrá una serie de gráficas para demostrar cómo evoluciona en el tiempo los precios.
Precondiciones	<ul style="list-style-type: none"> - El plugin debe de estar instalado en la web. - Deben de existir productos con el plugin ya activo.
Escenario principal	<ol style="list-style-type: none"> 1.El administrador inicia sesión en el panel de administración de WordPress. 2.Acceder al panel de configuración del plugin y acceder a un producto que ya tenga activo el cambio de precio. 3.El administrador introduce la configuración para el producto.
Escenario alternativo	No existe escenario alternativo.

Nombre	Descripción
Caso de uso	Importar casos reales.
Actores	Usuario administrador.
Descripción	Debemos de poder importar casos reales durante el periodo de testeo del algoritmo para poder ver su eficacia.
Precondiciones	Tener un dataset con en formato EXCEL con la ventas reales del videojuego.
Escenario principal	<ol style="list-style-type: none"> 1.El usuario Introduce la configuración de los posibles compradores y la

	configuración para el producto durante un periodo de tiempo. 2.Importa el dataset al sistema. 3.Observar los resultados por medio de gráficas para ver la comparativa.
Escenario alternativo	No existe escenario alternativo.

Nombre	Descripción
Caso de uso	Comprar en la web
Actores	Usuario estándar.
Descripción	Debe de ser posible para el usuario la compra de los productos en la web.
Precondiciones	Ninguna
Escenario principal	1.Entrar en la web. 2.Seleccionar los juegos a comprar e introducir la información para aplicar la compra del juego.
Escenario alternativo	No existe escenario alternativo.

Nombre	Descripción
Caso de uso	Realizar comentarios.
Actores	Usuario estándar.
Descripción	Los productos de la web pueden tener una lista de comentarios.
Precondiciones	Ninguna
Escenario principal	1.Entrar en la web. 2.Seleccionar el juego.

	3.Escribir un comentario.
Escenario alternativo	No existe escenario alternativo.

Nombre	Descripción
Caso de uso	Buscar producto.
Actores	Usuario estándar.
Descripción	la web debe de disponer de algún filtro para facilitar la busque de un producto en concreto.
Precondiciones	Ninguna
Escenario principal	1.Entrar en la web. 2.Seleccionar los juegos a comprar e introducir la información para aplicar la compra del juego.
Escenario alternativo	No existe escenario alternativo.

Nombre	Descripción
Caso de uso	CRUD de productos en la web.
Actores	Usuario administrador.
Descripción	La web debe de permitirnos la creación , lectura, actualización y eliminación de productos.
Precondiciones	Ninguna
Escenario principal	1.El administrador inicia sesión en el panel de administración de WordPress. 2.Acceder al panel de configuración del producto y aplicar la acción pertinente.

Escenario alternativo	No existe escenario alternativo.

3

Implementación de la web

3.1 Visión general

Como ya se comentó anteriormente, como parte del proyecto nos dispusimos a crear una web en WordPress para la implementación del algoritmo de venta de contenidos digitales.

A continuación explicaré las distintas partes que tuvo el desarrollo de la web y el análisis previo de donde implementar la web las limitaciones del sistema que tuvimos, ya que nos ceñimos a hosting gratuitos y ver cuáles eran las ventajas y desventajas que encontramos entre las web que pudimos encontrar.

Pero antes de enfocarnos directamente en la web vamos a explicar brevemente que es realmente WordPress y por qué entre otros motivos escogimos este CMS para el desarrollo del proyecto.

3.2 WordPress

WordPress es un sistema de gestión de contenidos web (CMS o content management system), que en pocas palabras es un sistema para publicar contenido en la web de forma sencilla. Su uso está realmente extendido siendo el líder a nivel mundial para la creación de webs desde hace bastante tiempo.

Es un software de código abierto gratuito cuyo objetivo es un sistema de gestión de contenidos que no dependa del personal puramente técnico, si no que cualquier persona pueda crear un sitio web propio. [1]

WordPress se divide en tres partes:

- Core: WordPress en sí, que es gratuito y descargable.

- Temas: Sirven para cambiar la apariencia de la web y existen un enorme catalogo tanto gratuitos como de pago.
- Plugins: Utilidades permiten mejorar tu web en casi cualquier cosa. Igualmente que los temas, hay un repositorio gratuito y miles de empresas que venden sus funcionalidades.

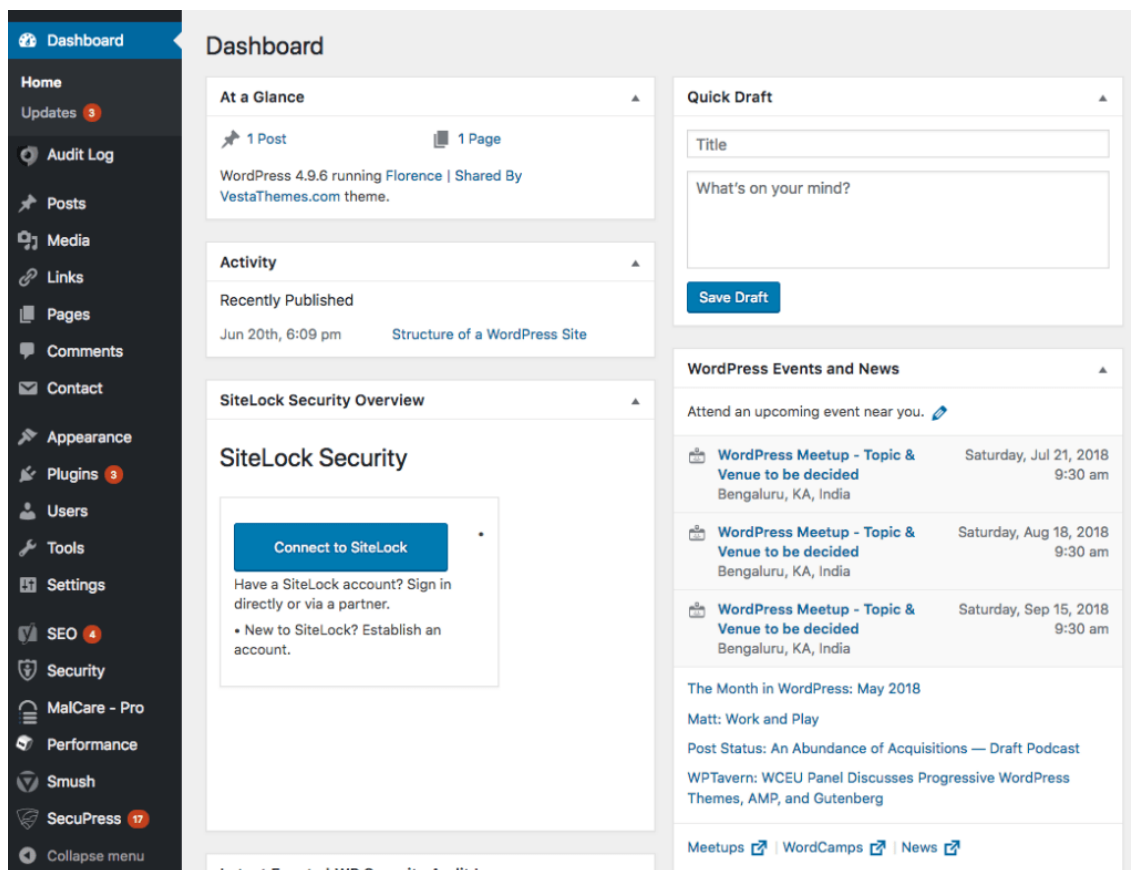
3.2.1 Estructura de WordPress

WordPress consta de un montón de archivos, y los principales son wp-admin, wp-content, wp-includes y wp-config. Estos se encuentran dentro de una carpeta llamada public_html (véase en la siguiente imagen):



3.2.2 wp-admin

La carpeta wp-admin contiene archivos para configurar el panel administrador de WordPress. Es decir, lo primero que ve después de iniciar sesión en su sitio de WordPress. Se parece a esto:



Este apartado es el encargado de verificar las credenciales, el tipo de usuario que está iniciando sesión o incluso habilitar funciones dentro del panel de administración de WordPress, establecer conexiones a la base de datos, ...

3.2.3 wp-includes

Dentro de la carpeta wp-includes encontramos los archivos responsables de el aspecto de la web basada en WordPress. Debido a esto, muchos de los archivos principales de WordPress se almacenan aquí.

3.2.4 wp-content

La carpeta wp-content es donde se almacenan los temas, plugins y otros elementos de la web (imágenes, pdf, ...). Siempre que instala un tema, se almacena en la carpeta “themes”, los plugins se instalan en la carpeta “plugins” y las imágenes que se usan en la web se almacenan en la carpeta “uploads”.

3.2.5 wp-config

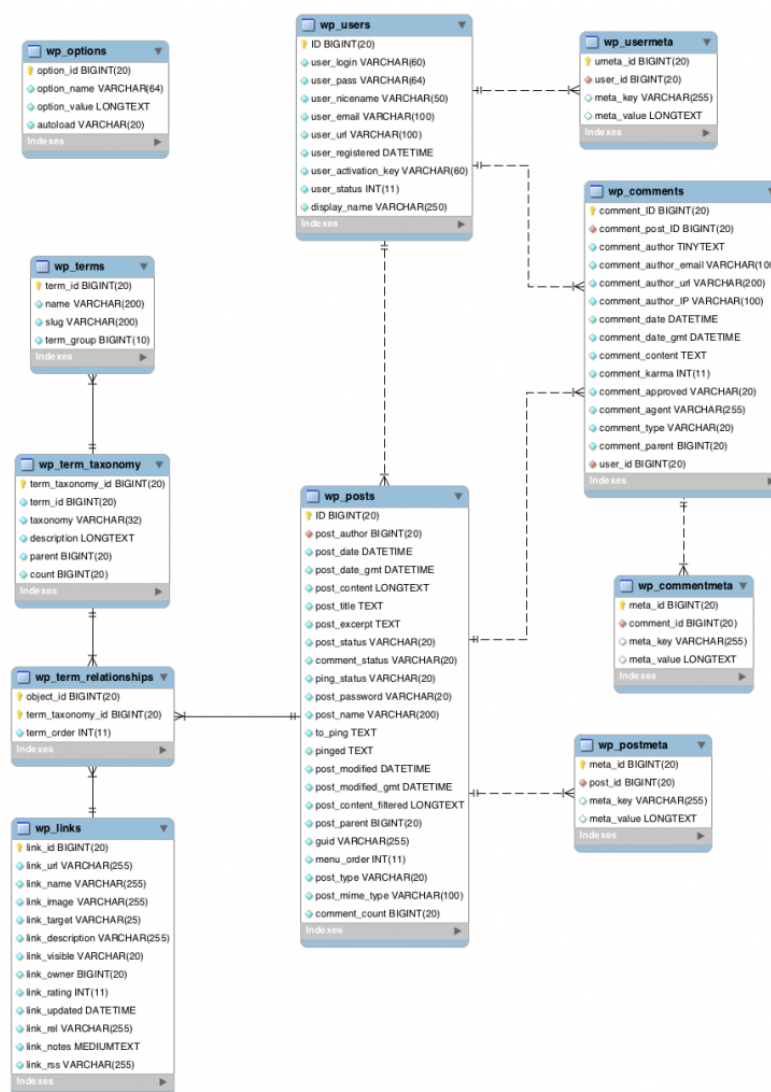
El archivo wp-config es el encargado de establecer la comunicación entre WordPress y la base de datos. Es un archivo extremadamente importante y no debe modificarse. Las modificaciones pueden romper la conexión entre la base

de datos y WordPress. Este archivo permite configurar además diversos parámetros que afectan al uso de WordPress.

3.2.6 Base de datos de WordPress

WordPress como toda web posee los que llamamos base de datos, por así decirlo podríamos definirlo como un armario con muchos estantes, que almacena archivos de forma estructurada. Por defecto, la base de datos tiene once tablas. Las mesas se pueden pensar en espacios de almacenamiento dentro de un armario. Las tablas se expanden a medida que comience a usar el sitio web de WordPress. Las tablas de la base de datos se caracterizan por tener un prefijo predeterminado **wp_** que se puede cambiar a voluntad. Estas tablas se pueden optimizar y reparar, y también se pueden exportar o hacer copias de seguridad con plugins como BlogVault.

En la siguiente imagen podemos apreciar cual es la estructura base de la base de datos de una página WordPress. [2]



No vamos a explicar en profundidad su funcionamiento pero si comentar por encima y que tablas se definen y para qué sirven.

- **wp_commentmeta** – Cualquier metadato asociado a los comentarios, como claves y valores, se almacenan en esta tabla. No contiene metadatos de autores o fechas, pero sí otros datos que WordPress utiliza ocasionalmente. Está íntimamente relacionada con la tabla wp_comments.
- **wp_comments** – Esta tabla contiene todos los comentarios que se hacen en entradas o páginas de WordPress, así como información relacionada como el autor, su dirección de correo electrónico y la entrada en la que se dejó el comentario.
- **wp_links** – Esta tabla se utiliza para almacenar la información de los enlaces del blogroll de WordPress. Aunque a partir de la versión 3.5 de WordPress desapareció el blogroll la tabla aún sigue ahí por si lo quieres activar.
- **wp_options** – Cualquier opción que hayas modificado en el panel de ajustes se almacena en esta tabla..
- **wp_postmeta** – Al igual que wp_commentmeta, wp_postmeta incluye información adicional y opcional, en este caso de las entradas. Salvo en temas que requieren almacenar shortcodes propios y cosas así es una tabla que no se usa mucho.
- **wp_posts** – wp_posts contiene toda la información de las entradas y sus datos asociados. Esta tabla es quizás la más utilizada pues contiene todo el contenido del sitio.
- **wp_terms** – Esta tabla almacena las etiquetas y categorías con las que se organizan las entradas.
- **wp_term_relationships** – Esta tabla es la responsable de mantener las relaciones entre las entradas y las etiquetas y categorías que tengan asociadas. También está relacionada con la tabla wp_term_taxonomy.
- **wp_term_taxonomy** – Esta tabla relaciona qué tipo de taxonomías se asocian a las entradas. Está relacionada con wp_terms y wp_term_relationships, registrando asociaciones entre categorías, etiquetas, etc.
- **wp_usermeta** – Otra tabla meta, wp_usermeta almacena cualquier metadato adicional y opcional asociado a la tabla wp_users.
- **wp_users** – wp_users contiene todos los datos de los usuarios registrados en WordPress, incluyendo información como los nombres de usuario, permisos y contraseñas.

Durante el desarrollo del plugin, que explicamos más adelante, modificamos un poco la estructura de la figura anterior con dos nuevas tablas que se usan de forma exclusiva para nuestro plugin.


3.3 Hosting

Un hosting se define como un sistema de almacenamiento en línea que nos permite publicar sitios o aplicaciones web en internet. Cuando obtenemos un hosting alquilamos un espacio dentro de un servidor para poder alojar ahí los archivos y datos de nuestra web o aplicación.

En internet se pueden encontrar varios sitios web donde poder alojar la webs para WordPress de forma gratuita pero existían diversas limitaciones al no tener que pagar por ello.

Estas son algunas de las soluciones que existen actualmente con sus limitaciones respectivamente:

	Ventajas
	<ul style="list-style-type: none">▪ 1 Sitio web▪ 300 MB de Almacenamiento HDD▪ ~300 Visitas Mensuales help_outline▪ Ancho de banda Limitado (3 GB)▪ WordPress Administrado▪ WordPress Optimizado
	<ul style="list-style-type: none">▪ 1000 MB de espacio SSD NVMe▪ 5000 MB de tráfico mensual▪ 1 cuenta de correo electrónico▪ Hosting con cPanel gratuito▪ Instalador de aplicaciones en un clic▪ Sin límite de bases de datos▪ Soporte técnico 24/7▪ Sin publicidad ni anuncios▪ Hosting gratis para WordPress: instálalo en 1 clic▪ Alta automática, tendrás acceso en segundos
	<ul style="list-style-type: none">▪ Dos sitios de zona de pruebas gratuitos▪ Entornos en vivo, de prueba y de desarrollo basados en Git▪ Certificados HTTPS (SSL) gratuitos▪ Red de entrega de contenido integrada (CDN)
	<ul style="list-style-type: none">▪ Cuentas FTP: 1/1▪ Subdominios: Ilimitado▪ Dominios complementarios: ilimitados▪ Dominios estacionados: Ilimitados▪ Bases de datos MySQL: ilimitado▪ Disco: ilimitada▪ Ancho de banda: ilimitado

	<ul style="list-style-type: none"> ▪ 1 GB de espacio en disco ▪ 5 GB de tráfico mensual ▪ Alojamiento de 1 dominio 3 subdominios ▪ Servidor DNS gratuito ▪ PHP, Perl / CGI-BIN 1 base de datos MySQL ▪ Acceso FTP 1 correo electrónico POP3 / IMAP ▪ Configuración de cuenta instantánea ▪ Administrador de archivos basado en web ▪ Panel de control fácil de usar
---	--

Después de evaluar diferentes opciones, como proveedor hosting seleccionamos pantheon.io, ya que este nos proporcionaba la mayor cantidad de las herramientas que necesitábamos de forma clara y concisa.

3.4 Plugins

Dentro del ecosistema de WordPress existe una enorme cantidad de plugins, usados para poder extender y mejorar las capacidades del CMS. Muchas de estas herramientas son gratuitas y pueden cambiar profundamente el funcionamiento de la web, ahorrándonos mucho trabajo en algunos casos.

Durante el desarrollo de este trabajo fueron usados los siguientes plugins externos (esto es, además del plugin propio para el algoritmo de cálculo de precio):



WooCommerce es una solución de código abierto. enfocado en el comercio online.

Este plugin contiene una gran cantidad de mejoras para las tiendas online. Algunas de las mejoras que posee son: temas para la venta de los productos, un mayor control de configuración de los productos para los administradores, Configuraciones de métodos de pago, ...



Themify es un proveedor de temas premium de WordPress, Contiene numerosos temas que puedes usar en tu web y además te da la posibilidad de crear los tuyos propios sin la necesidad de tocar el código.

El método de uso es sencillo solo tienes que arrastrar y soltar los módulos ya definidos los usuarios pueden construir sitios rápidamente, sin una línea de código.

Posee una gran cantidad de plantillas ya funcionales y es compatible con la gran mayoría de plugins. En nuestro caso usamos un tema y algunos plugins del mismo.

4

Diseño del algoritmo

4.1 Visión general

Para el desarrollo del proyecto se dieron varios entregables funcionales. Para lograr una mejor comprensión de cada uno de ellos explicaremos cada uno de ellos por separado: Un conjunto de data-set (que representan las ventas reales de algunos de los videojuegos del momento), un conjunto de prototipos del algoritmo (estos representan distintas versiones a la hora de evaluar), un programa de consola para testar el algoritmo seleccionado y finalmente un plugin con la implantación del algoritmo final.

4.2 Data-sets

Para poder hacer la comparativa de los resultados que proporcionarían los prototipos que íbamos a desarrollar, era necesario tener una estimación afinada de cuáles serían las ventas de los productos (videojuegos en este caso) que se obtienen al vender durante varios meses día a día, (esto sería necesario para poder ver cómo evoluciona la venta del algoritmo durante los primeros meses, que es el periodo de mayor venta en la mayoría de juegos), pero no era posible obtener lista detallada, ya que las empresas no suelen proporcionar esta información al público [4][5].

Por ello, los data-sets fueron elaborados por medio de estimaciones basadas en las distintas noticias que se podían encontrar en la web.

Los data-sets eran simples y solo constaban de:

- La fecha de la venta.
- El número de ventas.

- El precio que tenía el video juego durante aquel periodo.

4.3 Prototipos iniciales

Para el desarrollo del algoritmo, al no tener una idea definida a priori de cómo desarrollarlo, sino que sólo se contaba con una teoría intuitiva basada en la experiencia de como se comportan los clientes, se comenzó por definir una serie de posibles prototipos para el mismo. Todos ellos compartiendo como base el cálculo de la tendencia que tiene el producto agrupándolo por periodos.

Prototipo 1: El análisis se aplica de forma directa con el periodo anterior, mirando siempre si el último periodo fue positivo o negativo y en base a eso modificar el precio actual del producto.

Prototipo 2: Aquí nos basamos en hacer un análisis de tendencia de un determinado periodo de tiempo inicial, de forma que cuando el algoritmo detecte que la tendencia va a cambiar y por ejemplo, el juego va a dejar de ser comprado, se aplique un cambio de precio.

Prototipo 3: Análisis basado en el estudio de la tendencia de otros Juegos de las mismas características (categorías que compartan acción, lucha, shooter,) que consideremos exitosos, cambiando el precio cuando vemos que el volumen de ventas baja con respecto a los otros.

Algoritmo	Pros	Contras
Prototipo 1	Sencillo y fácil de aplicar.	<ul style="list-style-type: none"> - Demasiado directa cambia el precio una vez las ventas ya han disminuido. - No tiene en cuenta influencias externas.
Prototipo 2	Algo mas complejo que el caso anterior pero contiene una mayor base matemática detrás.	<ul style="list-style-type: none"> - No tiene en cuenta influencias externas.
Prototipo 3	Entendemos que los resultados son mejores que en los casos anteriores ya que nos basamos en juegos ya exitosos.	<ul style="list-style-type: none"> - Necesita de una base ya calculada de tendencias de otros videojuegos ya en el mercado.

Después de aplicar el análisis que vemos en la tabla superior descartamos el prototipo 3 por necesitar de una base de datos grande para poder funcionar. El objetivo final es implantarlo en un plugin que pueda llegar a ser comercial y buscamos que no se requiera de mucho tiempo o que dependa de datos externos. Por lo que finalmente entre los dos algoritmos restantes el prototipo 2 fue el seleccionado ya que proporcionaba mejores resultados que el prototipo 1.

4.4 Desarrollo de la aplicación de consola

Una vez seleccionado el prototipo del algoritmo que vamos a seguir, era necesario crear una aplicación previa al plugin que nos permitiese evaluar y testear los resultados de manera más cómoda y poder ajustar como los distintos compradores actuarían ante los cambios de precios. Por ello se decidió el desarrollo de una aplicación de consola en C# (este cambio en el lenguaje de programación se dio, para facilitar y acelerar el proceso de creación del algoritmo ya que es el lenguaje de programación en el que tengo más experiencia por el momento) y posteriormente se adaptaría el código a PHP para incluirlo dentro del plugin.

4.4.1 Funcionamiento del Algoritmo

El algoritmo seleccionado se basó en el cálculo de la tendencia de series temporales. Una serie temporal definirse como una sucesión de observaciones de una variable en distintos momentos del tiempo. Estudiando estas observaciones podemos identificar la tendencia y así poder predecir como la serie se va a comportar en el futuro [7].

Para nuestro caso, usamos el método de *las dos medias* para el cálculo de la tendencia, el cuál sigue la siguiente fórmula:

$$y = \frac{y_2 - y_1}{t_2 - t_1} \times (t - t_1) + y_1$$

Donde:

t_2	Media aritmética de las fechas del segundo periodo.
t_1	Media aritmética de las fechas del primer periodo.
y_1	Media aritmética de las ventas del primer periodo.
y_2	Media aritmética de las ventas del segundo periodo.

Para su correcto uso, necesitamos al menos tener información de 2 periodos temporales para así poder hacer los cálculos necesarios. En este caso, cada vez que se complete un periodo, se calcula la tendencia con respecto al pasado. En caso de que la tendencia sea descendiente, el precio se decrementará para así poder aumentar el número de ventas. Análogamente, se aumentará el precio en el caso contrario para así poder maximizar el beneficio.

4.4.2 Configuración y uso del algoritmo

La aplicación, para poder usar su funcionalidad al completo, necesita de un dataset con datos de venta reales o estimados del producto. Tales datos se usaran para comparar las ventas simuladas con las reales. También se permite ejecutar el algoritmo de manera simple para evaluar cual sería el mejor precio para la siguiente semana o calcular como se comportaría a lo largo de un tiempo determinado. Los datos calculados pueden ser exportados a Excel o bien ser visualizados en gráficas para su mejor comprensión.

Como se puede apreciar en la siguiente imagen, la configuración de la aplicación consta de 2 partes:

The screenshot shows the application's configuration interface. It is divided into two main sections: 'Algorithm configuration' (highlighted with a green border) and 'Possible buyer configuration' (highlighted with a red border). The 'Algorithm configuration' section includes fields for 'Product Price', 'Product Max Price', 'Product Min Price', 'Change Amount', and 'Period', along with an 'Include Extra configuration' checkbox and an 'Algorithm extra configuration' section with 'Change the prices in case the sales are very unexpective', 'Change Percentage', 'Period', and 'Start configuration after period'. The 'Possible buyer configuration' section includes fields for 'Total Buyers per day', '%Cautious buyers', '%Occasional buyers', '%Normal buyers', '%Compulsive buyers', '%Aggressive buyers', '%Random Buyers', and 'Months to simulate'. To the right of these sections are buttons for 'Simulate', 'Export', and 'Graph'. Below the 'Possible buyer configuration' section is a 'Real Data' section with 'Import CSV' and 'Real data file name' fields, and a 'Comparison Benefits' section with 'Calculate', 'Benefits- Real', and 'Benefit algorithm' fields.

Por un lado, aparece la configuración del propio algoritmo, representada por el recuadro **verde**. Esta configuración, a su vez, se subdivide en 2 apartados: la configuración base que ocupa la parte superior y la configuración extra en la parte inferior (debemos aclarar en este punto que la configuración extra no pudo ser completada debido a falta de tiempo y a la gran cantidad de trabajo invertida en la mejora del algoritmo). La configuración base consta de las siguientes variables:

- **Precio del producto:** Representa el precio base a partir del cual se calcula el nuevo precio.
- **Máximo valor del producto:** Representa la cifra máxima a la que puede llegar el precio del producto.
- **Mínimo valor del producto:** Representa la cifra mínima a la que puede llegar el precio del producto.
- **Capacidad de cambio:** Representa cuando puede variar el precio del Juego entre una iteración y otra.
- **Prioridad:** Cantidad de tiempo que se tiene que dar entre un cálculo de precio y el siguiente en días.

Por la otra parte, a la derecha, dentro del recuadro **rojo**, observamos que podemos introducir la configuración de los distintos compradores. Introducimos el número inicial de potenciales compradores (este número se irá reduciendo a medida que el tiempo avance y dichos compradores potenciales pasen a ser reales. Esto es, vayan comprando el juego) e indicamos que porcentaje de ellos corresponden a cada tipo de comprador.

Para poder emular de la forma más precisa las posibles acciones de los compradores, teníamos que identificar qué tipos de compradores existen y que caracteriza a cada uno de ellos. Para ello creamos 6 modelos distintos de compradores que se distinguen por su forma de actuar a la hora de comprar:

- **Compulsivo:** El comprador compulsivo sería la representación de los fanes de un producto y suelen ser de los primeros en adquirir el producto. y suelen comprar el juego tanto suba o baje el precio a partes iguales.
- **Normal:** Los compradores normales son aquellos que acaban comprando el Juego tras un tiempo de que aparezca el producto.
- **Cauto:** Este tipo de comprador es el que rara vez compra algo y de comprar algo solamente se da cuando hay una gran rebaja.
- **Agresivo:** Representa a aquellos compradores que valoran más el producto ante una subida de precio ya que lo valoraría como que el producto merece la pena. En el caso contrario es muy difícil que compre el producto porque esto le genera desconfianza.
- **Ocasional:** El comprador ocasional representa aquel que quizás compre el producto por que un amigo o familiar lo tiene y este se encuentra en oferta.
- **Aleatorio:** Estos son los compradores que aparecen de forma más aleatoria, no son compradores regulares si no que representan aquellos casos de compra que no son esperados.

Una vez definidos los tipos de compradores para poder de emular su comportamiento, se le aplicó una probabilidad a que cada tipo comprador para realizar la acción de comprar el producto. Para la programación de este apartado se emplearon principios y técnicas de ingeniería del software (por ejemplo principios SOLID como el Open-Close, usando polimorfismo y herencia de clases) con objeto garantizar la capacidad de evolucionar el algoritmo en el futuro.

4.4.3 Pruebas del algoritmo

Como ya se comentó en un apartado anterior la aplicación de consola es capaz de mostrarnos algunos graficas para así poder entender y observar los resultados del algoritmo.

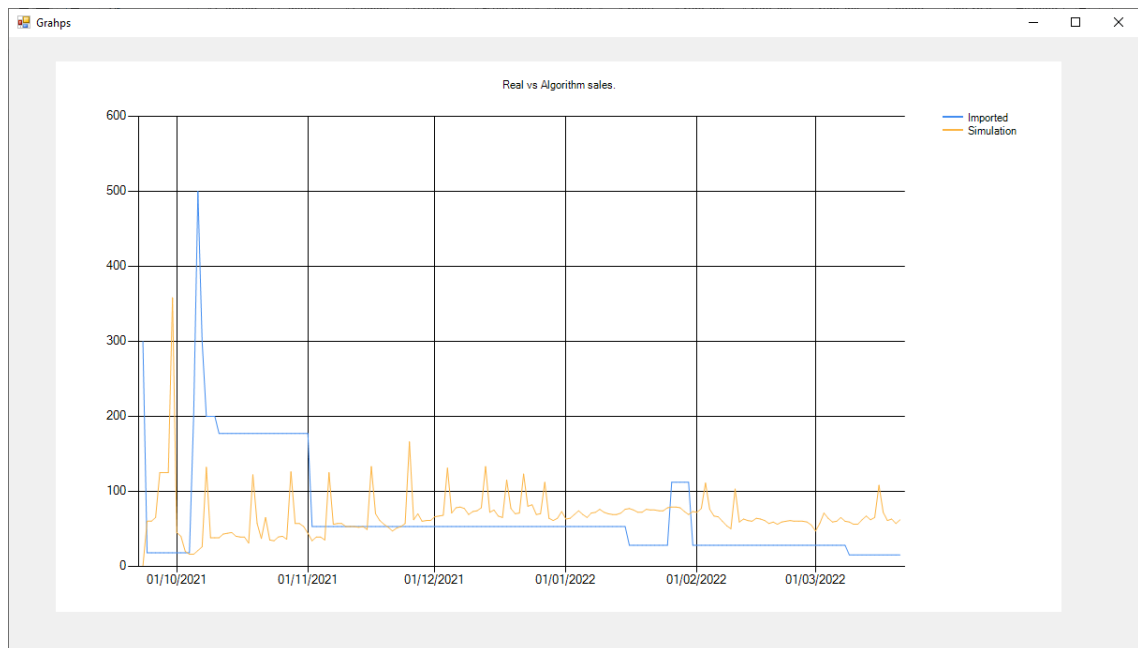
A continuación, observaremos como interactúan las ventas y precios con los compradores a lo largo del tiempo, en el caso hipotético de que full guys (como ejemplo de videojuego dentro de nuestra web) hubiese usado nuestro algoritmo [6].

Dada la siguiente configuración:

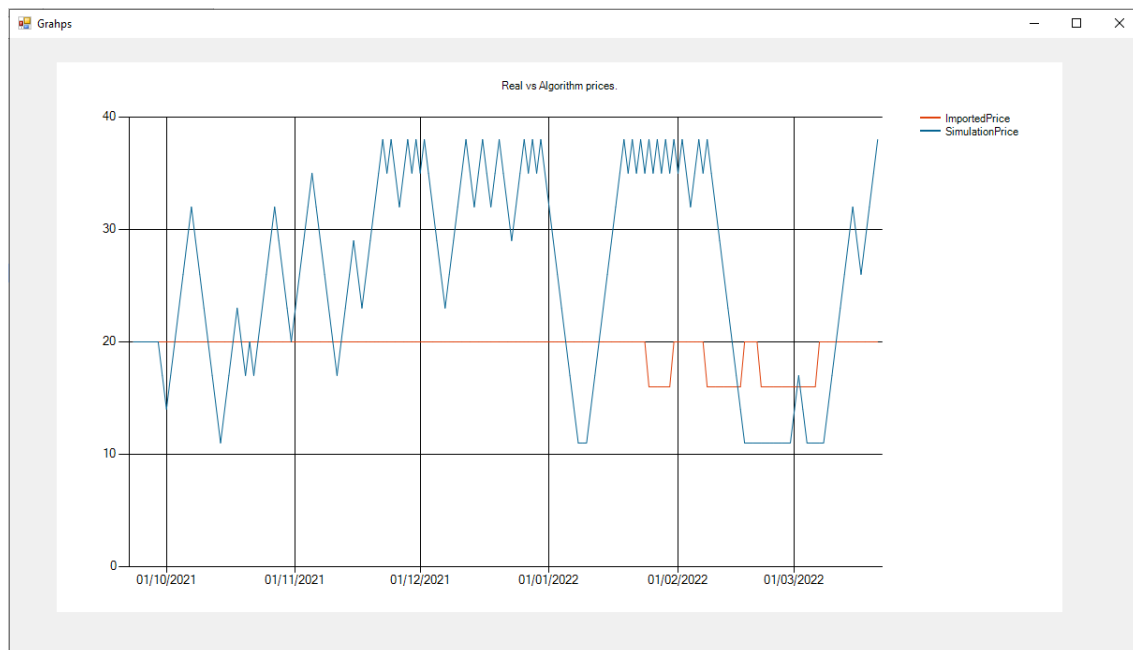
Algorithm configuration		Possible buyer configuration	
Product Price:	20	Total Buyers per day	30
Product Max Price:	40	%Cautious buyers:	20
Product Min Price:	10	%Occasional buyers:	10
Change Amount:	3	%Normal buyers:	40
Period:	7 Days	%Compulsive buyers:	15
<input type="checkbox"/> Include Extra configuration		%Aggressive buyers:	10
		%Random Buyers:	5
		Months to simulate:	6

Algorithm extra configuration

Estos serían los resultados obtenidos en relación al número de ventas:



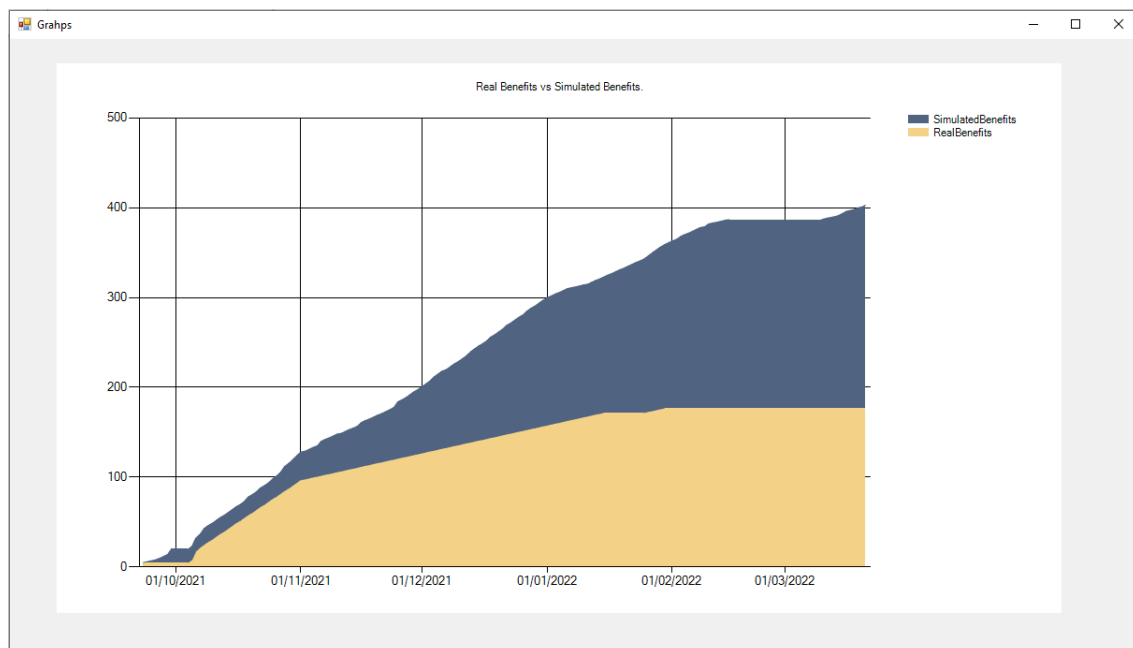
Estos serían los resultados en relación a como fluctúa el precio a lo largo del tiempo:



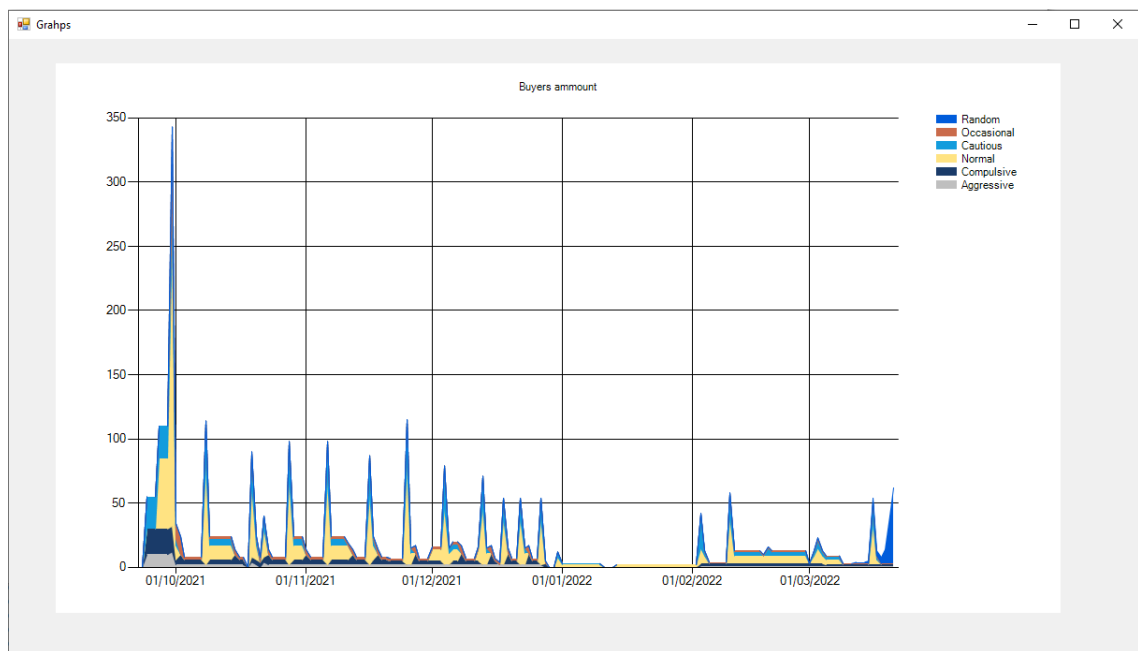
Mirando de forma general todas las tablas, podemos observar que el mayor volumen de ventas se da en un inicio (cuando tenemos más compradores) al igual que en el caso real, si bien es cierto que en el caso real el número absoluto de ventas de esta primera fase es ligeramente superior. Pero incluso teniendo menos ventas que el caso real, gracias a que el precio de nuestro producto se incrementa, obtenemos un beneficio mayor que en el caso real como puede verse en la siguiente imagen.

Como podemos apreciar, las ventas de ambos sistemas tienden a disminuir pero el caso de nuestro algoritmo mantiene más constante el volumen de compradores a lo largo del tiempo.

Aquí podemos apreciar los beneficios:



Estos serían los resultados en relación a como los compradores actúan:



Agresivos: Como era de esperar, los compradores agresivos compraron el producto durante las primeras semanas mientras el precio subía. Tras ese periodo, no volvemos a encontrar este tipo de compradores ya que su número es reducido.

Compulsivos: Al igual que los compradores agresivos, principalmente aparecen durante los primeros periodos de la venta, aunque también seguían apareciendo, si bien en un número más reducido, durante el resto del tiempo.

Normales: Los compradores normales se reparten a lo largo de todos los periodos y aparecen de forma constante.

Cautos: Estos compradores representan a aquellas personas que rara vez compran algo a menos que esté muy claro que es una ocasión, y como podía esperarse solo compraron el producto cuando estaba rebajado.

Ocasional: El comprador ocasional representa aquel que quizás compre el producto porque un amigo o familiar lo tiene y este se encuentra en oferta.

Aleatorio: Estos son los compradores que aparecen de forma más aleatoria, al igual que los compradores normales los encontramos a lo largo de todos los periodos pero de una forma más inconsistente.

Finalmente, la herramienta también nos permite hacer un cálculo de los beneficios que aproximadamente podrían haber obtenido por ambas partes y aunque la diferencia no es muy grande, se puede considerar significativa, demostrando que nuestro algoritmo mejora los beneficios.

Comparison Benefits

Calculate	Benefits- Real	278940
	Benefit algorithm	321698

4.5 Plugin.

Un plugin es un pequeño programa instalable desde la plataforma de WordPress para ampliar o extender las funcionalidades que WordPress trae por defecto. Existe una enorme cantidad de plugins gratuitos en el repositorio de WordPress (más de 50.000 y este número aumenta día a día), para hacer casi cualquier cosa con nuestra web [8].

Existen ciertas herramientas y estructuras que seguir para poder crear nuestro plugin. Ahora revelaremos cuales son los elementos mínimos para crear un plugin de WordPress y cual fue la estructura seguida para el desarrollo del nuestro.

A la hora de crear un plugin de WordPress podemos tener dos situaciones:

1. Podemos tener que la funcionalidad que vamos a añadirá a WordPress es realmente pequeña que no requiere de estructuras complejas como puede ser la ejecución de un código antes de loguearse en la web, podemos simplemente generar un archivo con extensión .php dentro de la carpeta /wp-content/plugins/ (El nombre del plugin será el nombre que hayamos proporcionado al archivo php).
2. Por otra parte, podemos necesitar una estructura bien formada de carpetas en caso de que nuestro plugin es más complejo.

Para el segundo caso existen múltiples formas de estructurar el proyecto en función de nuestras necesidades, pero un ejemplo de estructura genérico puede ser el que vemos en la siguiente imagen.

```

1 /nombre-plugin
2     nombre-plugin.php
3     uninstall.php
4     index.php
5     readme.txt
6     /languages
7     /includes
8     /admin
9
10    /js
11    /css
12    /images
13
14    /public
15        /js
16        /css
17        /images

```

- **‘nombre-plugin.php’**: Es el archivo principal del plugin. Contiene una cabecera estándar con los datos mínimos que WordPress necesita para realizar la instalación del plugin. Además, se definen las constantes y se realizan acciones iniciales.

- **‘uninstall.php’**: Es el fichero que define los métodos y acciones de limpieza para una correcta desinstalación del plugin.
- **‘index.php’**: Es un fichero PHP vacío con el único propósito de ocultar la estructura de carpetas. Si un usuario entra en la ruta de la carpeta del plugin se encontrará una página en blanco. Si se elimina aparecerá la estructura de carpetas.
- **‘readme.txt’**: Es un fichero con un formato concreto que se requiere solo si el plugin se va a subir al repositorio oficial de WordPress.
- **‘admin’**: Directorio que contiene subdirectorios con los archivos *.php, *.css, *.js, *.html que se encargan del back-end del plugin.
- **Directorio ‘languages’**: Directorio que contiene los archivos de internacionalización y localización del plugin.
- **Directorio ‘includes’**: Los archivos *.php auxiliares que “incluyas” desde la sentencia include de PHP a tu complemento.
- **Directorio ‘public’**: Directorio que contiene subdirectorios con los archivos *.php, *.css, *.js, *.html que se encargan del front-end del plugin.

4.5.1 Herramientas de WordPress

Para el desarrollo de un plugin de WordPress suele ser necesario principalmente el uso de Hooks y taxonomías:

- **Hooks**: los hooks o ganchos de WordPress funciones que te permiten añadir funciones en puntos concretos de la ejecución del código de WordPress.
- **Taxonomías**: La taxonomías son clasificaciones u ordenaciones en grupos de elementos que tienen características comunes. En WordPress utilizamos las taxonomías para agrupar y ordenar Post o Custom Post Types.

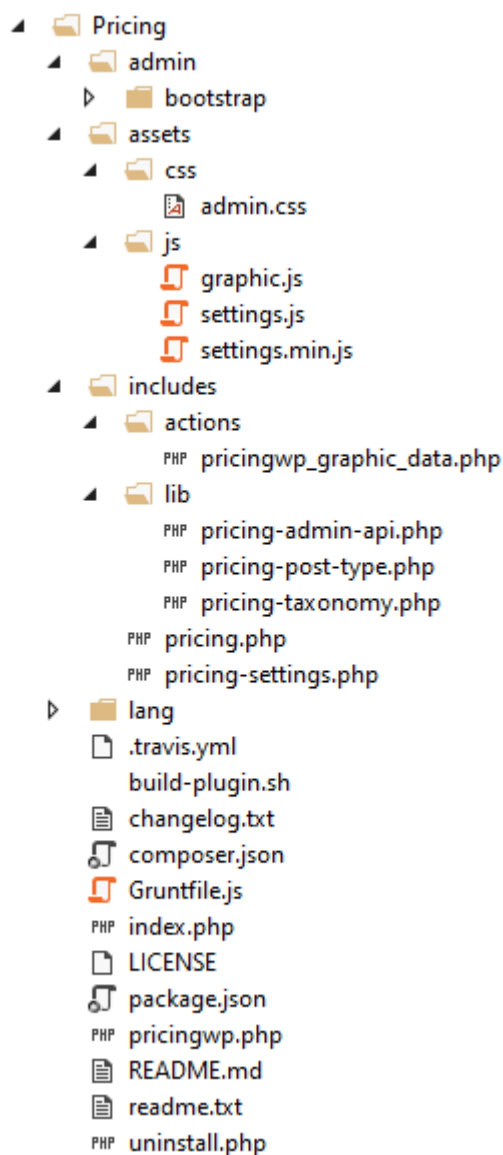
4.5.2 Desarrollo del Plugin

Para la creación del plugin de WordPress se hizo uso de una plantilla (template) ya existente proporcionado por la comunidad de WordPress en github. Este template consta de una estructura bien formada para el desarrollo del proyecto con una subdivisión de carpetas y archivos ya coherente [11].

Como finalmente se optó por la creación de eventos para calcular los precios de forma dinámica la mayor parte de la lógica se encuentra en pricing.php archivo donde se da la instalación del mismo.

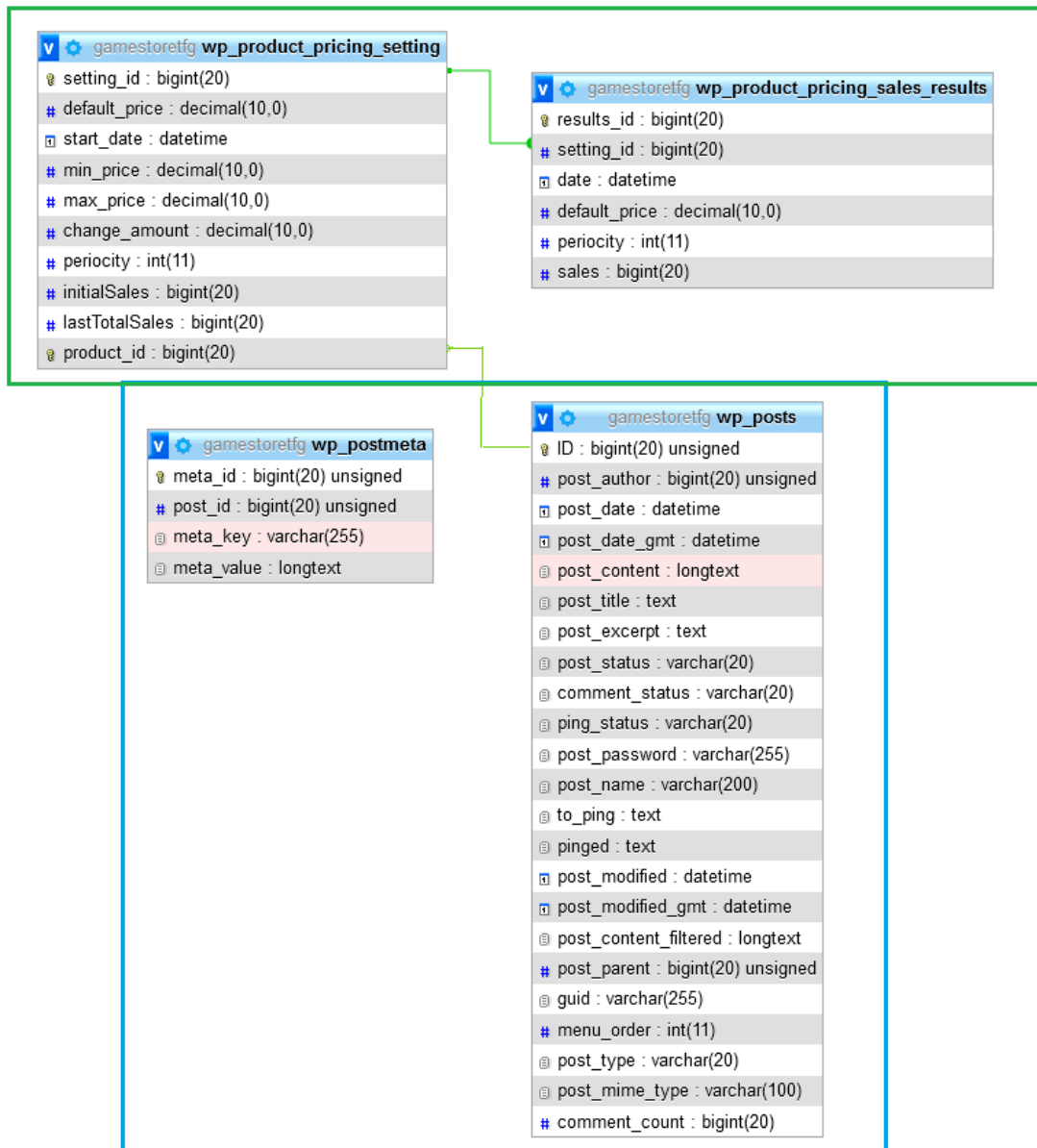
El archivo pricing-settings.php es el encargado de configurar y mostrar la página de administración para el administrador, esta se nutre de las demás para crear los distintos elementos y hace uso principalmente de pricing-admin-api.php, donde están los métodos para generar el html que muestra la página.

En la siguiente imagen podemos observar la estructura de carpetas usadas en el proyecto.



4.6 Base de datos del plugin .

Para el funcionamiento del plugin es necesario aplicar ciertas modificaciones al esquema de la base de datos. Por ello, se crearon 2 nuevas tablas *wp_product_pricing_setting* y *wp_product_pricing_sales_results* las cuales podemos ver representadas en la siguiente imagen junto con las otras dos tablas que son afectadas por el plugin *wp_posts* y *wp_postmeta*:



Como podemos apreciar en la imagen las tablas dentro del recuadro verde son las tablas creadas por el plugin y sus respectivas tareas son:

- *wp_product_pricing_setting*: Es la table encargada se guardar las configuraciones del plugin para los distintos productos y contiene la información mínima necesaria para la ejecución del algoritmo.
- *wp_product_pricing_sales_results*: Contiene un histórico con los distintos resultados obtenidos por el algoritmo. Esta table es principalmente usada para proveer los datos que mostramos en las gráficas del plugin.

Por otra parte las otras dos tablas mencionadas anteriormente *wp_posts* y *wp_postmeta*. Son utilizadas para Extraer ciertos datos del producto a configurar como el precio de los producto o el número total de ventas.

4.7 Front-end plugin

El diseño de la interfaz del plugin se limita al apartado de configuración dentro de la configuración de WordPress (ya que por la parte del comprador la única diferencia visual se encuentra en el precio del producto que se modifica a través de su valor en la base datos).

El plugin busca ser sencillo y fácil de usar de tal manera que se lo bastante intuitivo como para que sin guía sea claro el método de uso.

Como se puede apreciar en la siguiente imagen el plugin consta de una tabla donde podremos configurar el precio para cada uno de los productos de nuestra web.

Product Name	Current Price	Maximum Price	Minimum Price	Change Amount	Period	Actions
Zombie Army 4: Dead War	49.99	40	10	3	7	Deactivate Settings
Valheim	30	45	10	3	5	Deactivate Settings
Titanfall® 2	38.80					Activate
The Medium	49.99	20	10	3	7	Deactivate Settings
The Forest	33.99					Activate
test2	30					Activate
test	30	40	10	3	7	Deactivate Settings
Street Fighter V	7.99					Activate
Sea of Thieves	39.99					Activate
Red Dead Redemption 2	59.99					Activate

Dentro de cada uno de ellos una vez activamos el plugin pasamos a un apartado e configuración del producto, donde el administrador escoge su configuración única.

Pricing Settings

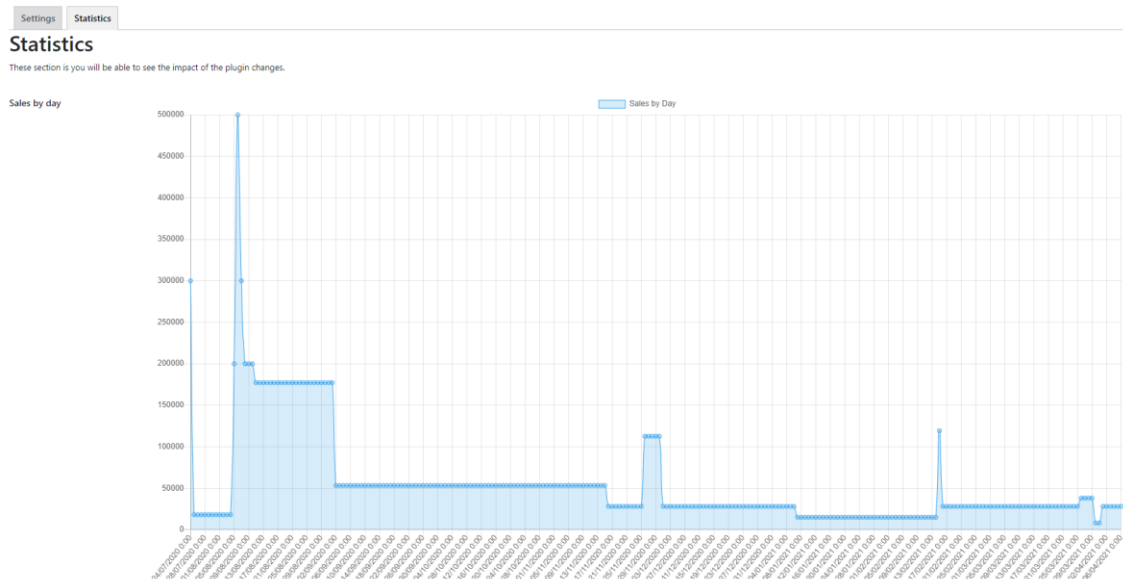
Settings Statistics

Settings

Here you can set a default configuration for your products, please fill all the elements.

Current price*	<input type="text" value="30"/>	This value can be modify in the product administration area.
Max Price*	<input type="text" value="40"/>	This is the maximun price set by default for all the games.
Min Price*	<input type="text" value="10"/>	This is the minimum price set by default for all the games.
Amount of change*	<input type="text" value="3"/>	This value represent how much the price will be increase or decrease depeding of the results get from the market after the periocity is complete.
Periocity*	<input type="text" value="7"/>	Number that will set how many days are needed to change the price.
<input type="button" value="Save Settings"/> <input type="button" value="Back"/>		

En el caso del producto tenga ya activa la configuración del precio se nos aparece un botón adicional para poder visualizar en gráficas como ha ido evolucionando el precio del producto en el tiempo y cuáles son los beneficios por periodo desde que la funcionalidad de cambio de precio se encuentra activa.



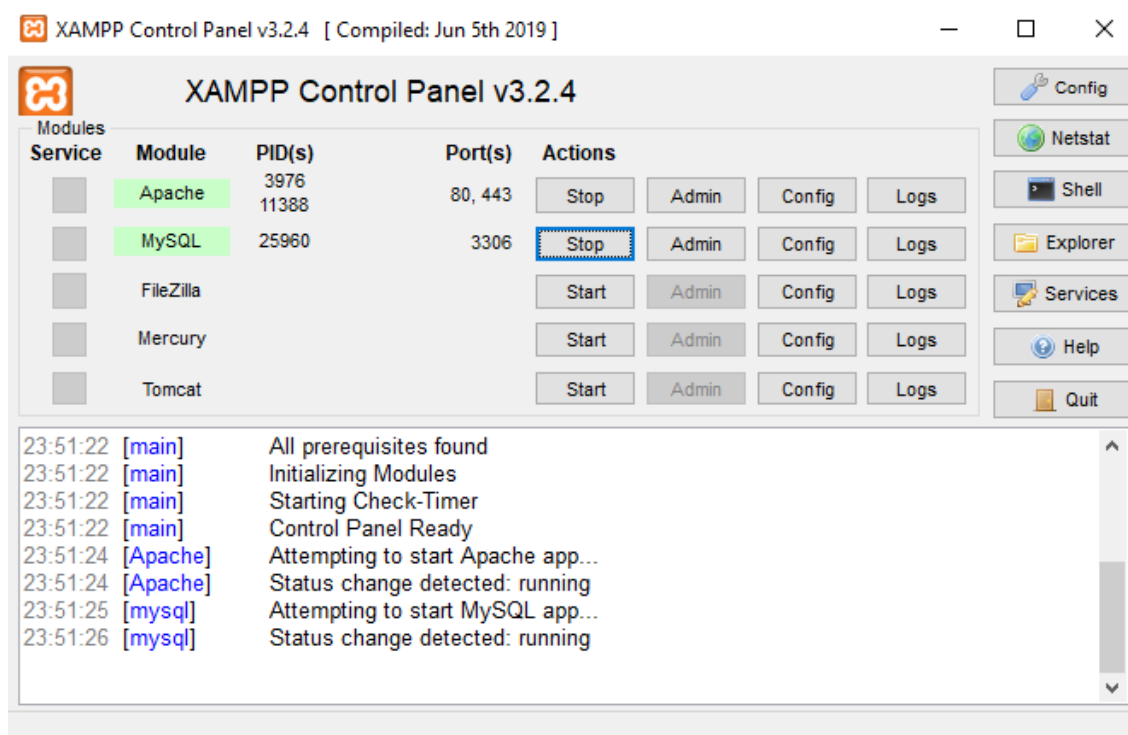
4.8 Instalación de un servidor local (XAMPP)

Para el correcto desarrollo del plugin era necesario emular un servidor local donde ejecutar un la web WordPress. Para esta tarea se usó la archiconocida herramienta XAMPP, una distribución de Apache que incluye varios componentes de software libres. El nombre proviene de un acrónimo compuesto por las iniciales de los principales componentes que usa: Apache, los sistemas bases de datos MySQL y MariaDB, así como Perl y PHP.[1]

- **Apache:** el servidor web de código abierto es la aplicación más usada globalmente para la entrega de contenidos web. Las aplicaciones del servidor son ofrecidas como software libre por la Apache Software Foundation.
- **MySQL/MariaDB:** conMySQL, XAMPP cuenta con uno de los sistemas relacionales de gestión de bases de datos más populares del mundo. En combinación con el servidor web Apache y el lenguaje PHP, MySQL sirve para el almacenamiento de datos para servicios web. En las versiones actuales de XAMPP esta base de datos se ha sustituido por MariaDB, una ramificación (“Fork”) del proyecto MySQL.
- **PHP:** es un lenguaje de programación de código de lado del servidor que permite crear páginas web o aplicaciones dinámicas. Es independiente de plataforma y soporta varios sistemas de bases de datos.
- **Perl:** este lenguaje de programación se usa en la administración del sistema, en el desarrollo web y en la programación de red. También permite programar aplicaciones web dinámicas.

A continuación, se enumeran los pasos para la creación de un servidor local y su instalación Wordpress en el mismo:

1. Descarga XAMPP desde la página oficial (<https://www.apachefriends.org/es/>),
2. Descarga Wordpress, desde su página web oficial (<https://es.wordpress.org/>).
3. Ejecuta el archivo descargado de la web de Xampp e instalar con la configuración por defecto.
4. Finalizada la instalación, ejecutar el programa y se comprueba que estén activadas las casillas de Apache y MySQL (véase la imagen inferior).



5. Se accede a la página principal de Xampp, indicando en el navegador: "http://localhost" y acceder al panel de Xampp.
6. Se selecciona la opción "phpMyAdmin" del menú de herramientas, al acceder se debe indicar el usuario y la contraseña configurados previamente.
7. Se crea una nueva base de datos por defecto.
8. Descomprimir el paquete de WordPress, en el directorio raíz de páginas web de Xampp, que en este caso es c:/xampp/htdocs/.
9. Se abre el archivo "wp-config-sample.php" con un editor de texto y se actualizan los datos de conexión a la base de datos que se ha creado previamente en MySQL: nombre de la base de datos, usuario y contraseña.
10. Se guarda el archivo modificado, con el nombre "wp-config.php".
11. Se accede a la página de instalación de WordPress, indicando en el navegador web: <http://localhost/wordpress/wp-admin/install.php>
12. Se introducen los datos solicitados, y se pulsa "Instalar WordPress".

13. En este punto, WordPress ya estará instalado en el servidor y mostrará una contraseña para el usuario “admin”.

4.9 Problemas durante el desarrollo

Durante el desarrollo un proyecto no es raro que aparezcan problemas y situaciones con distintas formas de solucionarse que en algunos casos acarrearán que el proyecto vaya más en una dirección o en otra, para solventar la situación que tenemos delante. Durante este proyecto también encontramos algunas de estas situaciones, por ejemplo se dio la situación de que el algoritmo debe de ser autónomo y automático. Una vez incluida la configuración sobre los productos no es necesario que el administrador del sistema ejecute ninguna acción para la ejecución del plugin. Para esto se vieron dos posibles soluciones la creación de un **Evento** en la base de datos o la creación en el servidor de un **Cron job**.

El **Evento** no es más que una tarea la cual se ejecuta de forma automática en un momento previamente programado. Pero el problema con esta solución es que no hay manera sencilla de evaluar una llamada desde MySQL a un código PHP, por lo que si queríamos continuar con esta propuesta era necesario traducir el código del algoritmo a procedimientos SQL para así poder ejecutarlo desde la base de datos. Esta solución es poco recomendable ya que la lógica de negocio debe de alojarse dentro del proyecto web por patrón de diseño.

El **Cron job** es una utilidad de Linux que establece un comando o secuencia de comandos de horarios en tu servidor para que algunas tareas se ejecuten automáticamente a una hora y fecha especificadas. Una tarea cron es la tarea programada en sí. esta solución a diferencia de la anterior no necesita de mover la lógica del plugin a ningún área pero necesita de que este instalado en el servidor la librería ssh2 para que se pueda crear dinámicamente desde el código PHP.

Finalmente la opción escogida fue el uso de **Eventos** ya que no nos genera la dependencia de instalar y configurar la librería ssh2 en el servidor del cliente, a pesar de vernos obligados a mover la lógica a la base de datos [10].

5

Conclusiones

5.1 Conclusiones

A modo de conclusión, podríamos decir que los objetivos principales del proyecto se han cumplido: Se ha desarrollado, estudio y caracterizado de la evolución de precios de contenidos digitales y hemos creado un algoritmo discreto que calcula el mejor precio. El aprendizaje las siguientes tecnologías: PHP, JS, MYSQL, Reac, C#. Se ha creado un plugin de WordPress. Todas estas tecnologías suponen una gran mejora competitiva para mi futuro profesional, en el mundo laboral, y entre ellas destacaría el desarrollo del plugins de WordPress, que por lo que tengo entendido es un qué poca gente posee. Por lo que no descarto en un futuro seguir con esta línea de trabajo.

Respecto a mi valoración personal, este Trabajo de Fin de Grado me ha permitido valorar la importancia de un buen análisis, y mejorar mis aptitudes en el desarrollo de una página web, así como también he podido ampliar mis conocimientos sobre CMS, hosting e implementación web. También me ha enseñado a estimar y precisar los tiempos estableciendo las prioridades en un proceso de creación y desarrollo web.

A pesar de los problemas surgidos durante el desarrollo, pienso que el resultado final del proyecto cumple con mis expectativas, implementa todo lo aprendido durante el proceso. Me encuentro satisfecho con el resultado obtenido y con todo lo aprendido y me ha servido para aplicar todo lo aprendido desde que entre en la universidad.

5.2 Futuras mejoras y líneas de investigación

En opinión de este autor y del director de este trabajo, el algoritmo y el plugin que lo implementa todavía poseen un gran potencial de mejora. Por el momento, cubrimos las necesidades básicas que podríamos pedir a un sistema de estas características, pero una vez conseguida esta base, que sirve como prueba de

concepto, se podrían mejorar varios aspectos y desarrollar nuevas funcionalidades.

Algunos ejemplos podrían ser:

- Para la herramienta de consola se podría incluir un módulo que tenga en cuenta la categoría de juego y se fije en webs para poder establecer mejor como afecta ese parámetro a la venta del videojuego.
- Se podría hacer el algoritmo más configurable y que evalúe el precio en periodos inferiores de tiempo, es decir, que por ejemplo él que pueda cambiar a partir de una fecha en concreto o que sea modificable en subperiodos o que existan eventos que hagan cambiar el precio del juego antes de que el periodo se complete. Esto abriría otras posibilidades a la hora de optimizar y evaluar los resultados.
- Otra propuesta podría ser incluir un mayor número de gráficas a la parte administrativa del plugin, en las que podamos extraer predicciones de las ventas que se obtendrán en el futuro del producto que estemos vendiendo aplicando la misma teoría que usamos en la aplicación de consola.
- El desarrollo de un algoritmo parecido pero baso en algoritmos evolutivos.

Esas podrían ser algunas de las posibles líneas a seguir para mejorar tanto el algoritmo como el plugin, y en el caso de incluirlas, podrían configurarse como partes de una versión premium del plugin pudiendo así cobrar por él.

Referencias

- [1] Estructura de archivos en WordPress: <https://www.malcare.com/blog/beginners-guide-to-understanding-the-structure-of-a-wordpress-site/>
- [2] Cómo WordPress utiliza MySQL: <https://ayudawp.com/como-wordpress-utiliza-mysql/>
- [3] XAMP: <https://www.ionos.es/digitalguide/servidores/herramientas/instala-tu-servidor-local-xampp-en-unos-pocos-pasos/>
- [4] Herramienta de steam para el análisis de jugadores: <https://store.steampowered.com/stats/Steam-Game-and-Player-Statistics?l=english>
- [5] Noticias de Juegos para la elaboración de los data-sets: <https://vandal.elespanol.com/noticia/1350739642/fall-guys-ha-vendido-mas-de-10-millones-de-copias-solo-en-steam/>
<https://www.eurogamer.es/articles/2020-08-10-fall-guys-ha-vendido-mas-de-2-millones-de-copias-en-steam>
- [6] Modeling Human Behavior: <https://www.youtube.com/watch?v=m-Lr3FHkttc>
- [7] Estimación de la tendencia de una serie temporal:: http://davidespinoza.es/joomla/index.php?option=com_content&view=article&id=339:estimacion-de-la-tendencia-de-una-serie-temporal&catid=80:analisis-externo
- [8] Crear plugin de WordPress: <https://raiolanetworks.es/blog/crear-plugin-wordpress/>
- [9] Template para el plugin de WordPress: <https://github.com/hlashbrooke/WordPress-Plugin-Template>
- [10] Crear y configurar un cron job desde PHP: <https://code.tutsplus.com/tutorials/managing-cron-jobs-with-php--net-19428>
- [11] Template para el plugin: <https://github.com/hlashbrooke/WordPress-Plugin-Template>
- [12] Estructura de un proyecto WordPress: <https://www.malcare.com/blog/beginners-guide-to-understanding-the-structure-of-a-wordpress-site/>
- [13] WordPress: web: <https://dev-gstoregame.pantheonsite.io/>

6 Apéndice A

Manual de Usuario e Instalación

6.1 Introducción

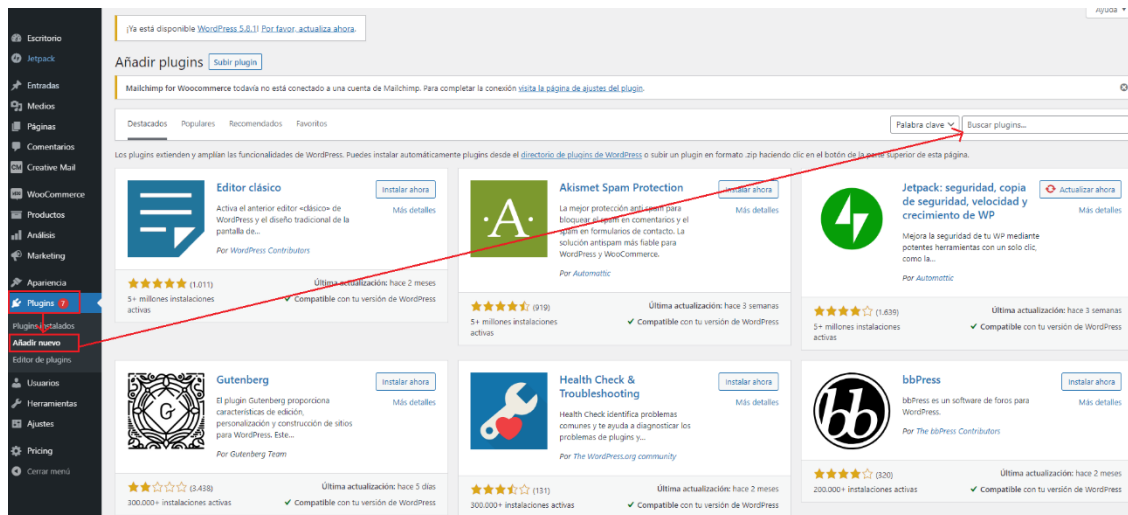
En este apéndice recopilamos los aspectos de instalación del proyecto que aparecen dispersos en diferentes sitios de la memoria, con objeto de facilitar su uso.

6.2 Instalación

El plugin no se encuentra alojado realmente en el repositorio de WordPress por lo que no es accesible desde la tienda aquí emularemos la acción de instalarlo como si fuera otro plugin.

Para la instalación del plugin solo tendremos que:

1. Acceder al panel de administración de WordPress
2. Seleccionar el apartado Plugins.
3. Añadir nuevo plugin.
4. Buscar el plugin (en nuestro caso el nombre es Pricing).



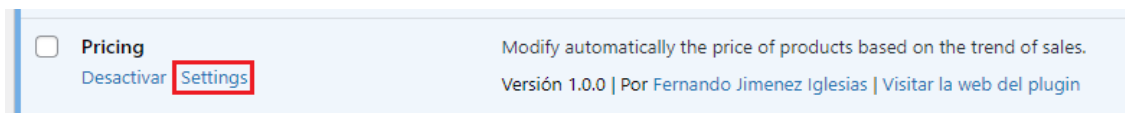
5. Una vez encontramos el plugin solo debemos de hacer click en instalar.



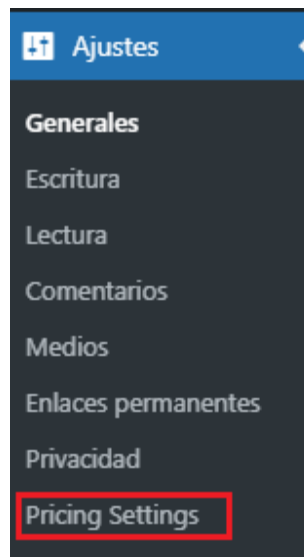
Nota: En nuestro caso al no encontrarse el plugin dentro de WordPress solo debemos de copar la carpeta con el proyecto dentro de \wp-content\plugins\.

6.3 Uso del plugin

1. Ahora podemos acceder de nuevo a la pestaña de plugins para activarlo.
2. Una vez activado para acceder al plugin solo tenemos que buscarlos en el apartado de plugin y hacer click en settings.



3. Alternativamente, también se puede acceder desde ajustes.



Nota: Aunque es posible, el plugin no fue añadido a como una de las opciones principales dentro del menú principal, ya que se considera que su uso es reducido. Una vez configurado no se espera que el usuario acceda a él de forma diaria.

4. Al acceder al plugin podemos ver un listado de los actuales productos dentro del sistema de WordPress.
5. Seleccionamos la opción de activar el plugin para introducir la configuración de cada producto.

(Ya está disponible [WordPress 5.8.1](#) por favor actualízalo ahora.)

Pricing Settings

Product Name:

Product Name	Current Price	Maximum Price	Minimum Price	Change Amount	Period	Actions
Zombie Army 4: Dead War	49.99	40	10	3	7	<input type="button" value="Deactivate"/> <input type="button" value="Settings"/>
Valheim	30	45	10	3	5	<input type="button" value="Deactivate"/> <input type="button" value="Settings"/>
Titanfall® 2	38.80					<input type="button" value="Activate"/>
The Medium	49.99	20	10	3	7	<input type="button" value="Deactivate"/> <input type="button" value="Settings"/>
The Forest	33.99					<input type="button" value="Activate"/>
test2	30					<input type="button" value="Activate"/>
test	30	40	10	3	7	<input type="button" value="Deactivate"/> <input type="button" value="Settings"/>
Street Fighter V	7.99					<input type="button" value="Activate"/>
Sea of Thieves	39.99					<input type="button" value="Activate"/>
Red Dead Redemption 2	59.99					<input type="button" value="Activate"/>

6. Tras hacer click en activar podremos acceder a un formulario donde introducir la configuración para nuestro producto.

Settings

Statistics

Settings

Here you can set a default configuration for your products, please fill all the elements.

Current price*

33.99

This value can be modify in the product administration area.

Max Price*

40

This is the maximum price set by default for all the games.

Min Price*

10

This is the minimum price set by default for all the games.

Amount of change*

3

This value represent how much the price will be increase or decrease depeding of the results get from the market after the periocity is complete.

Periocity*

7

Number that will set how many days are needed to change the price.

Save Settings

Back

- Una vez guardada la configuración se nos redirecciona a la tabla inicial donde podemos ver que la configuración a sido guardada.
- Si hacemos click en settings dentro de alguno de los productos activos podemos ver la configuración que introdujimos al activarlo y modificarla si así deseamos.
- En la parte superior dentro del menú de settings podemos observar un botón para acceder a las estadísticas que nos proporciona el sistema para ver cómo evolucionan las ventas y precios de cada producto.

